PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

SC
#11
5/20/03

Applicant	: Charles R. Musick et al.		Docket No.: IL-10443
Serial No.	: 09/479,432		Art Unit: 2172
Filed	: January 7, 2000		Examiner: A. Ly
For	: System And Method For Integrating And Accessing Multiple Data Sources Within A Data Warehouse Architecture		

BRIEF ON APPEAL

TABLE OF CONTENTS PAGE

This is an appeal to the Board of Patent Appeals and Interferences from the final rejection of Claims 1-53 mailed December 13, 2002. On March 13, 2003, a timely Notice of Appeal was filed.

I. REAL PARTIES IN INTEREST

The real parties in interest are the Regents of the University of California and the United States of America as represented by the United States Department of Energy.

II. RELATED APPEALS AND INTERFERENCES

Appellant knows of no other appeals or interferences that will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-53 are pending on appeal and stand rejected. A copy of the claims on appeal are set forth in Appendix I.

IV. STATUS OF AMENDMENTS

All prior amendments have been entered. An amendment canceling claims 32-40, 45 and 46 is attached hereto. These claims are redundant with claims 9-17, 22 and 23.

V. SUMMARY OF INVENTION

The present invention is a system and method for integrating and accessing multiple data sources within a data warehouse architecture. The system and method of the present invention are particularly advantageous over the prior art because a set of metadata is formed, providing a way to declaratively present domain specific knowledge, obtained by analyzing data sources, in a consistent and useable way. Four types of information are represented by the metadata: abstract concepts, databases descriptions, transformations and mappings.

Also, a mediator generator automatically generates data management computer code based on the metadata. The resulting code defines a translation library and a mediator class. The translation library provides a data representation for domain specific knowledge represented in a data warehouse, including "get" and "set" methods for attributes that call transformation methods and derive a value of an attribute if it is missing. The mediator class defines methods that take "distinguished" high-level objects as input and traverse their data structures and enter information into the data warehouse.

The invention includes a method for maintaining a data warehouse, including the steps of identifying a data source of interest, updating metadata to reflect information available from the source, automatically generating a mediator based on the metadata and writing a wrapper for the source which calls the mediator. A data warehouse is defined to be any code system for integrating multiple data sources, regardless of whether the approach is based on federated database, multidatabase, or traditional warehousing technology, and independent of the computer-useable medium on which the code is stored. Metadata is defined to be equivalent to ontology. The step of updating metadata includes entering new types of information, new data formats for previously defined information, new transformations between data formats, and the schema of the source. A stand-alone mediator generation program automatically generates a fully functional mediator. An API and translation libraries are automatically defined by the mediator generation program. The wrapper makes use of the mediator. The mediator may comprise code to translate between source and target representations, possibly using externally defined methods, and load data into the warehouse. The wrapper uses the API and public data structures defined by the mediator generation program. The mediator transforms and loads data into the warehouse.

The DataFoundry metadata model includes abstractions, translations, mappings and database descriptions. The model is described by a UML DataFoundry metadata representation, wherein the model defines the metadata used by a mediator generation program. The mediator generation program includes the steps of reading the metadata; generating translation libraries; generating an API; reading the metadata; and generating said mediator. Reading the metadata includes the steps of reading the abstraction metadata; reading the translation metadata; reading the database description metadata; and reading the mapping metadata. Translation libraries are generated by developing public and private class definitions and implementations of data structures, where the data structures comprise the abstractions and the translations.

Generating the mediator consists of creating public and private definitions and implementations of a class or classes capable of receiving data in one format, converting it to another format, and loading it into a data warehouse. Data is received by a receiving data structure defined within the translation library and is loaded into a warehouse whose schema corresponds to the database description component of the metadata. The method may be applied to a number of applications including data warehousing applications in the domain of protein sequence and structure analysis, data warehousing applications in the domain of functional genomics and proteomics, integrating a new data source into a data warehouse and updating a warehouse when a previously integrated data source is modified.

VI. ISSUES

Whether claim 9 is unpatentable over Knutson et al.

Whether claims 1-8, 11-21, 23-31, 41-44 and 47-53 are unpatentable over Knutson et al. in view of DeGroot et al.

Whether claims 10 and 22 are unpatentable over Knutson et al. in view of Fontana et al.

VII. GROUPING OF CLAIMS

Claims 1-31, 41-44 and 47-53 stand or fall together.

VIII. ARGUMENT

Is claim 9 unpatentable over Knutson et al.?

Four types of information are represented by the metadata of the present invention: abstract concepts, databases descriptions, transformations and mappings. See page 3, lines 16-17 of the application. Specifically, the applicants' metadata as recited in claim 9 includes translations between two databases through an abstract construct. The translation is carried out with a mediator, which is a program that translates data between two formats. See page 3, line 8 through page 4, line 6. Knutson's approach uses metadata within a data warehouse and as the basis for generating reports and does not include translations between two databases through an abstract construct. The reference does not include the metadata as claimed by the applicants. Therefore the rejection should be withdrawn.

Are claims 1-8, 11-21, 23-31, 41-44 and 47-53 unpatentable over Knutson et al. in view of DeGroot et al.?

Independent claims 1, 9, 24 and 47 all include the elements of a metadata and a mediator. Neither reference discloses these elements. Therefore the rejection should be withdrawn.

Are claims 10 and 22 unpatentable over Knutson et al. in view of Fontana et al.?

Claim 9 should be allowable over Knutson et al. as discussed above. Claims 10 and 22 depend from claim 9. Therefore the rejection should be withdrawn.

Accordingly it is submitted that the rejection of Claims 1-53 under 35 U.S.C. § 103(a) is improper and should be reversed.

Respectfully submitted,

John P. Wooldridge

John P. Wooldridge
Attorney for Appellants
Registration No. 38,725
Tel. No. (925) 456-2279

Dated: May 13, 2003

IX APPENDIX I

1. A method for maintaining a data warehouse, comprising:

identifying a data source of interest;

updating a metadata to reflect information available from said source;

automatically generating a mediator based on said metadata; and

writing a wrapper for said source which calls said mediator.

2. The method of claim 1, wherein the step of updating a metadata comprises entering new types of information, new data formats for previously defined information, new transformations between data formats, and the schema of said source.

3. The method of claim 1, wherein said mediator is fully functional and is automatically generated by a stand-alone mediator generation program.

4. The method of claim 3, wherein said mediator generation program automatically defines an API and translation libraries

5. The method of claim 4, wherein said mediator comprises code to translate between source and target representations, possibly using externally defined methods, and load data into said warehouse.

6. The method of claim 1, wherein said wrapper makes use of said mediator.

7. The method of claim 3, wherein said mediator generation program defines a public data representation, wherein said wrapper uses said public data representation.

8. The method of claim 3, wherein said wrapper uses said mediator to load data into said warehouse.

9. A DataFoundry metadata model comprising abstractions, translations, mappings and database descriptions.

10. The model of claim 9, comprising a UML DataFoundry metadata representation.

11. The model of claim 9, wherein said model defines metadata used by a mediator generation program, wherein said mediator generation program generates a mediator.

12. The model of claim 11, wherein said mediator generation program comprises:

reading said metadata;

generating translation libraries;

generating an API; and

generating said mediator.

13. The model of claim 12, wherein the step of reading said metadata comprises reading the abstraction metadata; reading the translation metadata; reading the database description metadata; and reading the mapping metadata.

14. The model of claim 12, wherein the step of generating translation libraries comprises developing public and private class definitions and implementations of data structures.

15. The model of claim 14, wherein said data structures comprise said abstractions and said translations.

16. The model of claim 12, wherein generating the mediator consists of creating public and private definitions and implementations of a class or classes capable of receiving data in one format, converting it to another format, and loading it into a data warehouse.

17. The model of claim 16, wherein said data is received by a receiving data structure defined within said translation library and said data is loaded into a warehouse whose schema corresponds to the database description component of the metadata.

18. The method of claim 1, wherein said method is applied to data warehousing applications in the domain of protein sequence and structure analysis.

19. The method of claim 1, wherein said method is applied to data warehousing applications in the domain of functional genomics and proteomics.

20. The method of claim 1, wherein said method is used for integrating a new data source into a data warehouse.

21. The method of claim 1, wherein said method is used for updating a warehouse when a previously integrated data source is modified.

22. The model of claim 9, as defined by the UML DataFoundry representation.

23. The method of claim 14, wherein said data structures correspond to said abstractions and said translations.

24. A computer-useable medium embodying computer program code for maintaining a data warehouse by executing the steps of:

identifying a data source of interest;

updating a metadata to reflect information available from said source;

automatically generating a mediator based on said metadata; and

writing a wrapper for said source which calls said mediator.

25. The computer-useable medium of claim 24, wherein the step of updating a metadata comprises entering new types of information, new data formats for previously defined information, new transformations between data formats, and the schema of said source.

26. The computer-useable medium of claim 24, wherein said mediator is fully functional and is automatically generated by a stand-alone mediator generation program.

27. The computer-useable medium of claim 24, wherein said mediator generation program automatically defines an API and translation libraries

28. The computer-useable medium of claim 27, wherein said mediator comprises code to translate between source and target representations, possibly using externally defined methods, and load data into said warehouse.

29. The computer-useable medium of claim 24, wherein said wrapper makes use of said mediator.

30. The computer-useable medium of claim 26, wherein said mediator generation program defines a public data representation, wherein said wrapper uses said public data representation.

31. The computer-useable medium of claim 26, wherein said wrapper uses said mediator to load data into said warehouse.

41. The computer-usable medium of claim 24, wherein said method is applied to data warehousing applications in the domain of protein sequence and structure analysis.

42. The computer-usable medium of claim 24, wherein said method is applied to data warehousing applications in the domain of functional genomics and proteomics.

43. The computer-usable medium of claim 24, wherein said method is used for integrating a new data source into a data warehouse.

44. The computer-usable medium of claim 24, wherein said method is used for updating a warehouse when a previously integrated data source is modified.

47. An apparatus for maintaining a data warehouse, comprising:

means for identifying a data source of interest;

means for updating a metadata to reflect information available from said source;

means for automatically generating a mediator based on said metadata; and

means for writing a wrapper for said source which calls said mediator.

48. The method of claim 1, wherein said method is applied to data warehousing applications in the domain of astrophysics and climate modeling.

49. The method of claim 1, wherein said method is applied to data warehousing applications in the domain of medical image processing and analysis.

50. The method of claim 1, wherein said method is applied to data warehousing applications in the domain of tracking consumer and customer preferences.

51. The method of claim 1, wherein said method is applied to data warehousing applications in the domain of satellite and terrestial communication systems analysis.

52. The method of claim 1, wherein said method is used for integrating a new data source into a data warehouse.

53. The method of claim 1, wherein said method is used for updating a warehouse when a previously integrated data source is modified.